ecs goodie implementation

[Note that the actual objects used do not have to be ecs objects!!]

A) structures
   i) HASH table
      - keyed by unique name & points to c-list & contains an open count
   ii) C-list
      - contains current object for ecs goodie
   iii) local open list in each process (contains local open count)

B) representation in directory
   - a unique name + type field (1 word)

C) ecs goodie capability
   - contains (in data udi) a unique name + type field

D) operations

I) creation of ecs goodie in directory
   - present a type field (default) (along with other data info for creating a directory entry)
   - creates an entry with a new unique name field, and type field that presented

II) get an ecs goodie from directory (may be uniform operation for all objects)
   - returns an ecs capability with:
     Type = "ecs goodie"
     options = Those indicated by the access key
     data = unique name + type field
III) Open eis goodie with a session object

presets eis goodie capability
object capability (destroy option? must be)

error if at least: i) proper bit not on in eis goodie option bits.
or ii) type field in data at of eis goodie capability
does not match that of presented object.

Then the eis goodie unique name field is looked up in the Hash table.
if found in the Hash table, F-return.
if not, make an entry in the hash table for this unique name,
point it to a free position in the C-list, place a copy
of the presented object capability at this location in
the C-list, and set open count in the Hash table
entry to 1. (also enter in local process open list)
and set local open count to 1.

If actual EIS object
object moved system alive lock changed as fixed space
c-list index

If found,
- compare
  - reference count
  - associated CEPS
  - CEPS size
- if equal, return
- else, continue
- create new entry
- add to CEPS list
- update local list
- return

If not found,
- F-return

Notably, CEPS returned is associated to the CEPS of the given CEPS entry.

Check local process list, if in the local list, return.
- If not in the local list, execute as follows:
  - If the local process list is empty, exit.
  - If the local process list is not empty, remove from local process list and execute as follows:
    - Check if the local process list is empty, if yes, return.
    - Else, continue.

Open CEPS (no object presented)

Join unique lower of data at object CEPS entry.

Present CEPS (no object presented)

Join unique lower of data at object CEPS entry.

Close object as follows:
- Present CEPS (no object presented)
- Open CEPS (no object presented)
- Join unique lower of data at object CEPS entry.